# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

am

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/975,864 | 10/11/2001 | Williams Ludwell Harrison III | 042390.P12861 | 9810 |

8791 7590 05/19/2005

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

| EXAMINER | |
|---|---|
| STEELMAN, MARY J | |
| **ART UNIT** | **PAPER NUMBER** |
| 2191 | |

DATE MAILED: 05/19/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

| Office Action Summary | Application No. | Applicant(s) |
|---|---|---|
| | 09/975,864 | HARRISON ET AL. |
| | **Examiner** | **Art Unit** |
| | Mary J. Steelman | 2191 |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *20 December 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *20 December 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☒ Other: *Copy of accepted drawings.*

## DETAILED ACTION

1. Per Applicant' request, the Specification has been amended. Claims 1, 2, 4, 5, 7-9, 11, 12, 14-16, 18, 19, 21-23, 25, 26 and 28 have been amended. Claims 1-28 are pending.

### *Drawings*

2. In view of the amendments to the drawings, the prior objections are hereby withdrawn.

### *Requirement for Information – 37 CFR 1.105*

3. Examiner acknowledges the receipt of MMX technology developers guide.

### *Claim Rejections - 35 USC § 112*

4. In view of the amendments to claims, the prior 35 USC 112 second paragraph rejection is hereby withdrawn.

### *Claim Rejections - 35 USC § 103*

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over MMX ™ CODE DEVELOPMENT STRATEGY (hereinafter referred to as MMX), in view of "Bitwidth Analysis with Application to Silicon Compilation", by Mark Stephenson, Jonathan Babb, and Saman Amarasinghe (ACM 2000), hereinafter referred to as 'Stephenson'

Per claims 1, 8, 15 and 22:

-determining…whether an operation on a larger data type maybe replaced by an operation on a smaller data type having a reduced precision…;

(MMX CODE DEVELOPMENT STRATEGY disclosed, in section 4.2, "Step one: Determine which code to convert." "It is these routines that will yield the greatest performance increase when converted to the MMX ™ technology optimized libraries code. Encapsulating these loops into MMX technology-optimized libraries will allow greater flexibility in supporting platforms with and without MMX technology." A performance optimization tool…may be used to isolate the compute-intensive sections of code. Once identified, an evaluation should be done to determine whether the current algorithm or a modified one will give the best performance (determining when an operation may be replaced). In some cases, it is possible to improve performance by changing the types of operations in the algorithm. Matching the algorithms to MMX technology instruction capabilities is key to extracting the best performance."

Section 4.3 "Is the Code Floating-Point or Integer" addresses the limitations of claims 1 and 2. "Step two: Determine whether the algorithm contains floating-point or integer data. If the current algorithm is implemented with integer data…If the algorithm contains floating-point data, then determine why floating point was used. Several reasons exist for using floating-point operations: performance, range and precision. If performance was the reason for implementing the algorithm in floating-point, then the algorithm is a candidate for conversion to MMX technology instructions to increase performance. If range or precision was an issue when implementing the algorithm in floating point then further investigation needs to be made. Can the data values be converted to integer with the required range and precision? If not, this code is best left as floating-point code."

Determine why a floating point was used, and whether an integer type (replace with a smaller data type) would be suitable)

-replacing the operation on the larger data type by the operation on the smaller data type as determined.

(MMX CODE DEVELOPMENT STRATEGY disclosed, Section 4.3 "Is the Code Floating-Point or Integer" addresses the limitations of claims 1 and 2. "Step two: Determine whether the algorithm contains floating-point or integer data. If the current algorithm is implemented with integer data...If the algorithm contains floating-point data, then determine why floating point was used. Several reasons exist for using floating-point operations: performance, range and precision. If performance was the reason for implementing the algorithm in floating-point, then the algorithm is a candidate for conversion to MMX technology instructions to increase performance. If range or precision was an issue when implementing the algorithm in floating point then further investigation needs to be made. Can the data values be converted to integer with the required range and precision? If not, this code is best left as floating-point code." Make a candidate for MMX technology, demote to integer type.)

MMX CODE DEVELOPMENT STRATEGY fails to disclose:

-determining, based on the analyzed code...

-wherein the operation on the larger data type is contained in code associated with a language implementation system...

-analyzing code associated with a language implementation system using bitwise constant propagation;

However, Stephenson disclosed (p. 108, first paragraph), "This paper introduces Bitwise, a compiler (analyzes code to determine replacement strategy)
0that minimizes the bitwidth (a language implementation system)...Because loop instructions comprise the bulk of dynamically executed instructions, Bitwise incorporates sophisticated loop analysis techniques (analyzing code / constant propagation) for identifying bitwidths", (p. 109, left column, Section 1.3), "The scope of Bitwise includes fixed-point arithmetic, bit manipulation (bitwise), and Boolean operations", (p. 109, left column, Section 1.5), "We introduce a suite of bitwidth extraction techniques that seamlessly perform bi-directional propagation (constant propagation)...We formulate an algorithm to accurately find bitwidth information in the presence of loops (constant propagation) by calculating closed-form solutions..."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to have used bitwise constant propagation when interpreting the code as disclosed by Stephenson because it may be used to (Stephenson: Abstract) "reduce silicon real estate" and "reduce power" of a system. Stephenson suggested (p. 108, Section 1.1), "Three new compilation targets for high-level languages are re-invigorating the need to conserve bits. Each of these architectures expose subword control. The first is the rejuvenation of SIMD architectures for multimedia workload. These architectures include Intel's MultiMedia eXtension (MMX)..." Thus Stephenson disclosed that bitwise constant propagation on MMX

architectures was suitable for optimizing performance by reducing bit widths through the use of a 'language implementation system (Bitwise compiler).

Per claims 2, 9, 16, and 23:

-determining whether a first variable of the larger data type may be replaced by a second variable of a smaller data type having the reduced precision;

(MMX: Section 4.3 "Is the Code Floating-Point or Integer" addresses the limitations of claims 1 and 2. "Step two: Determine whether the algorithm contains floating-point or integer data. If the current algorithm is implemented with integer data...If the algorithm contains floating-point data, then determine why floating point was used..." Determine why a floating point was used, may it be replaced?)

-replacing the first variable of the larger data type by the second variable of the smaller data type as determined.

(MMX CODE DEVELOPMENT STRATEGY disclosed, Section 4.3 "Is the Code Floating-Point or Integer" If performance was the reason for implementing the algorithm in floating-point, then the algorithm is a candidate for conversion to MMX technology instructions to increase performance. If range or precision was an issue when implementing the algorithm in floating point then further investigation needs to be made. Can the data values be converted to integer with the required range and precision? (replace with smaller data type) If not, this code is best left as floating-point code." Make a candidate for MMX technology, demote to integer type.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to have used bitwise constant propagation when interpreting the code as disclosed by Stephenson because it may be used to (Stephenson: Abstract) "reduce silicon real estate" and "reduce power" of a system. Stephenson suggested (p. 108, Section 1.1), "Three new compilation targets for high-level languages are re-invigorating the need to conserve bits. Each of these architectures expose subword control. The first is the rejuvenation of SIMD architectures for multimedia workload. These architectures include Intel's MultiMedia eXtension (MMX)..." Thus Stephenson disclosed that bitwise constant propagation on MMX architectures was suitable for optimizing performance by reducing bit widths through the use of a 'language implementation system (Bitwise compiler).

Per claims 3, 10, 17, and 24:

-replacing the operation and replacing the first variable are used for automatic vectorization for signal and media processors that provide vector operations on small fixed-point data types.

(MMX ™ CODE DEVELOPMENT STRATEGY, Section 4.7: MMX technology uses an SIMD technique to exploit the inherent parallelism of many multimedia algorithms (signal and media processors)...data should be formatted in memory according to the guidelines below...Converting this routine to MMX (replacing the operation and replacing the first variable) technology code, you would expect a four times speedup since MMX technology instruction can process four elements of the vector (vectorization) at a time using the MOVQ instruction, and perform four additions at a time using the PADDW instruction.)

Per claims 4, 5, 11, 12, 18, 19, 25, and 26:

MMX failed to specifically disclose:

- the processors are equipped to provide MMX / SSE instructions.

MMX and SSE™ are instruction sets that have been added to existing architectures. MMX instructions are SIMD for integers. SSE™ instructions (Streaming SIMD Extensions) are SIMD for single precision floating point numbers. Both are used for multimedia processing.

However, Stephenson disclosed that a 'Bitwise compiler' was suitable for MMX architectures (p. 108, Section 1.1, first paragraph).

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to have used a "Bitwise compiler" on an MMX architecture as disclosed by Stephenson because it may be used to (Stephenson: Abstract) "reduce silicon real estate" and "reduce power" of a system. Stephenson suggested (p. 108, Section 1.1), "Three new compilation targets for high-level languages are re-invigorating the need to conserve bits. Each of these architectures expose subword control. The first is the rejuvenation of SIMD architectures for multimedia workload. These architectures include Intel's MultiMedia eXtension (MMX)..." (Similarly, SSE instructions, streaming SIMD extensions would be obvious.) Thus Stephenson disclosed that bitwise constant propagation on MMX architectures was suitable for optimizing performance by reducing bit widths through the use of a 'language implementation system (Bitwise compiler).

Per claims 6, 13, and 27:

- performing algebraic simplification on the code.

(MMX ™ CODE DEVELOPMENT STRATEGY, Section 4.7: "An FIR filter is effectively a vector dot product (algebraic simplification on the code) in the length of the number of coefficient taps…" )

Per claims 7, 14, 21, and 28:

MMX disclosed:

Chapter 4 MMX ™ CODE DEVELOPMENT STRATEGY" disclosed code development strategies, including whether to demote code to make use of instruction set extensions.

The above mentioned article failed to disclose bitwise constant propagation. However Stephenson disclosed:

-the language implementation system performs the bitwise constant propagation by abstract interpretation on the code.

Stephenson: (p. 110, Section 3), "This section describes the infrastructure and algorithms of Bitwise, a compiler that perform bitwidth analysis. Bitwise uses SSA as its intermediate form (abstract interpretation on the code)…", (p. 110, Section 3.1, Propagating data-ranges, Figure 2(c)), "Technically, this representation maps bitwidth analysis to the more general value range propagation problem. Value range propagation is known to be useful in value predictions…constant propagation…"

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to have used bitwise constant propagation when interpreting the code as disclosed by Stephenson because it may be used to (Stephenson: Abstract) "reduce silicon real estate" and "reduce power" of a system. Stephenson suggested (p. 108, Section 1.1), "Three new compilation targets for high-level languages are re-invigorating the need to conserve bits. Each of these architectures expose subword control. The first is the rejuvenation of SIMD architectures for multimedia workload. These architectures include Intel's MultiMedia eXtension (MMX)..." Thus Stephenson disclosed that bitwise constant propagation on MMX architectures was suitable for optimizing performance.

Per claim 20:

- performing algebraic simplification on the code.

(MMX ™ CODE DEVELOPMENT STRATEGY, Section 4.7: "An FIR filter is effectively a vector dot product (algebraic simplification on the code) in the length of the number of coefficient taps..." )

***Response to Arguments***

7. Applicant's arguments with respect to claims 1-28 have been considered but are moot in view of the new grounds of rejection.

## *Conclusion*

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action. ***Conclusion***

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached at (571) 272-3694. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman
04/28/2005

TUAN DAM
SUPERVISORY PATENT EXAMINER

Accepted by Examiner 4-11-05 MV





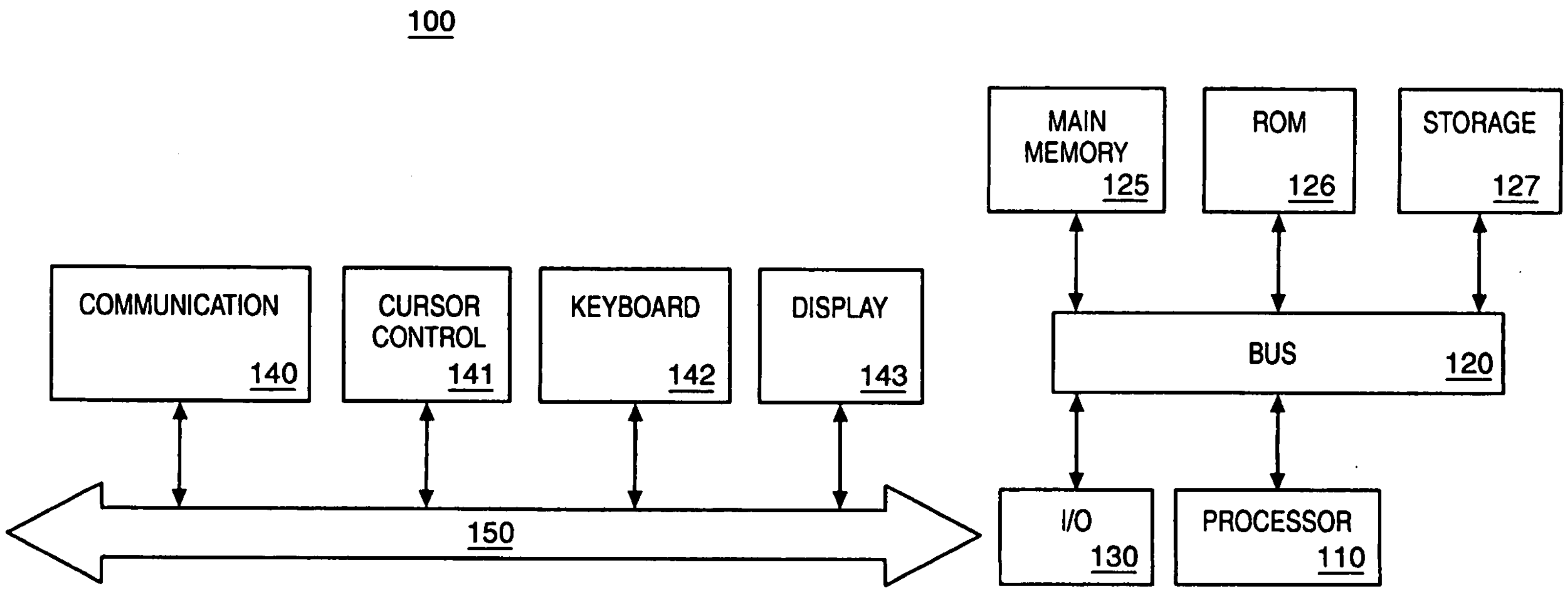


FIG. 1

Accepted by Examiner 4-11-05 MS





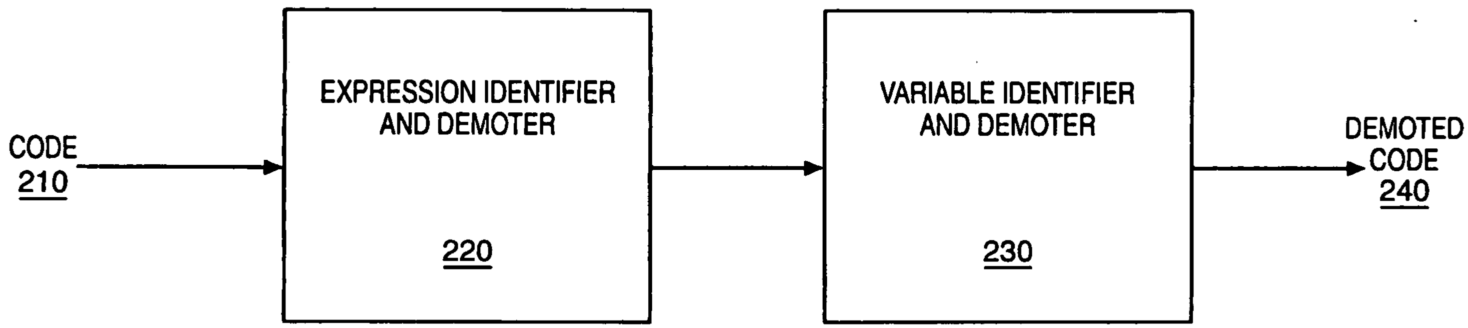


FIG. 2

Accepted by Examiner 4-18-05 MS



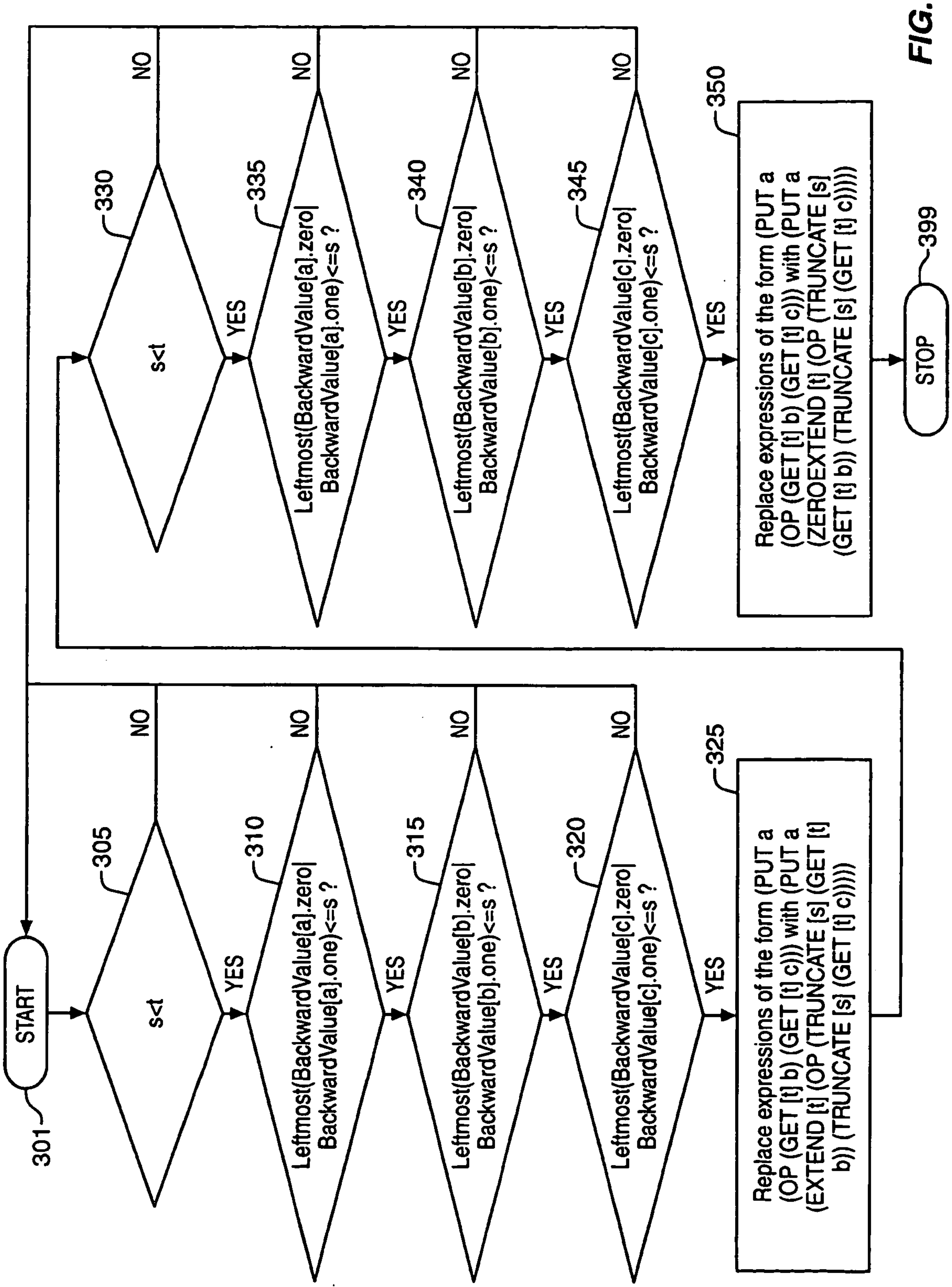


FIG. 3

Accepted by Examiner 4-11-05 MJ



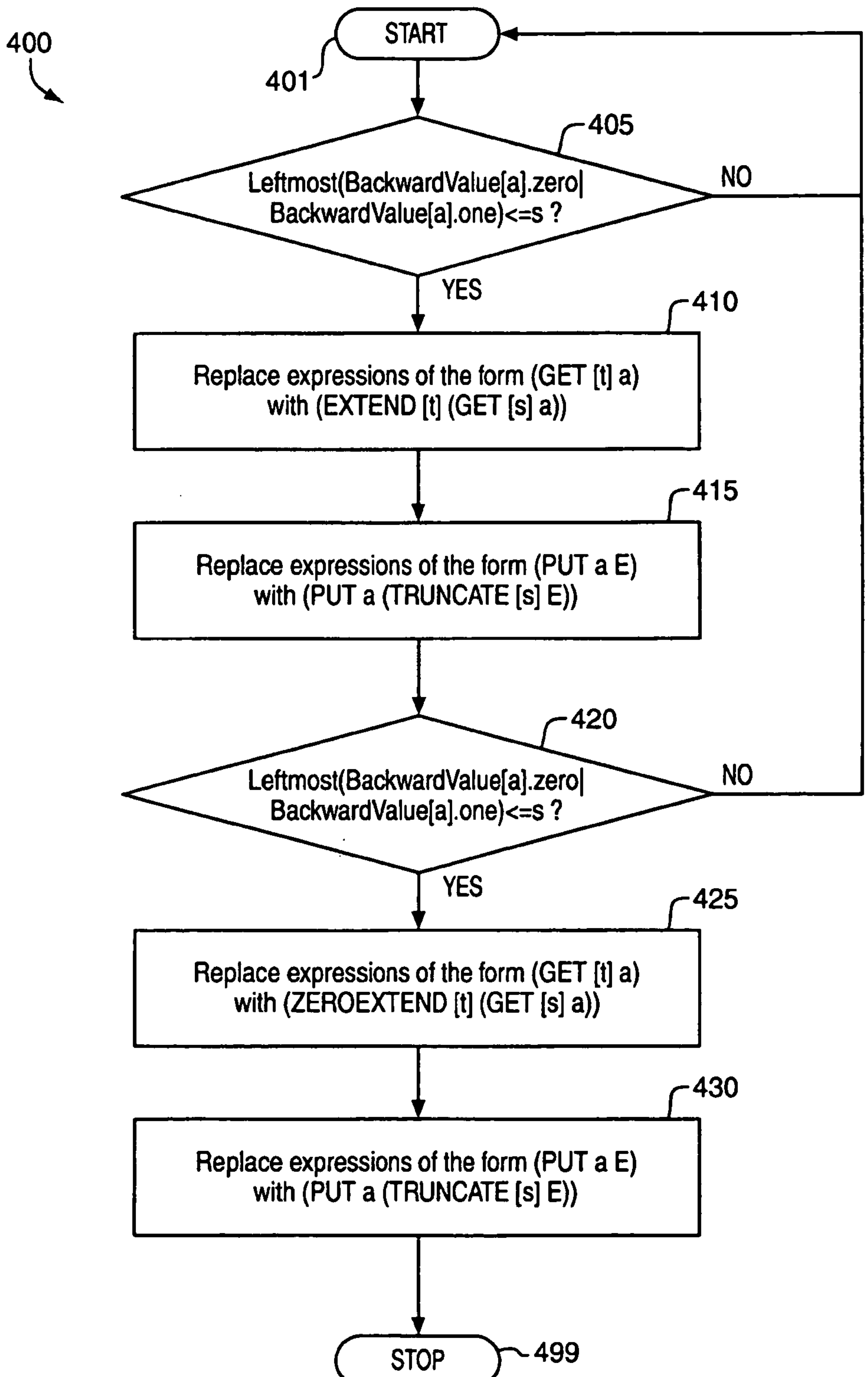


FIG. 4

Accepted by Examiner 4-11-05



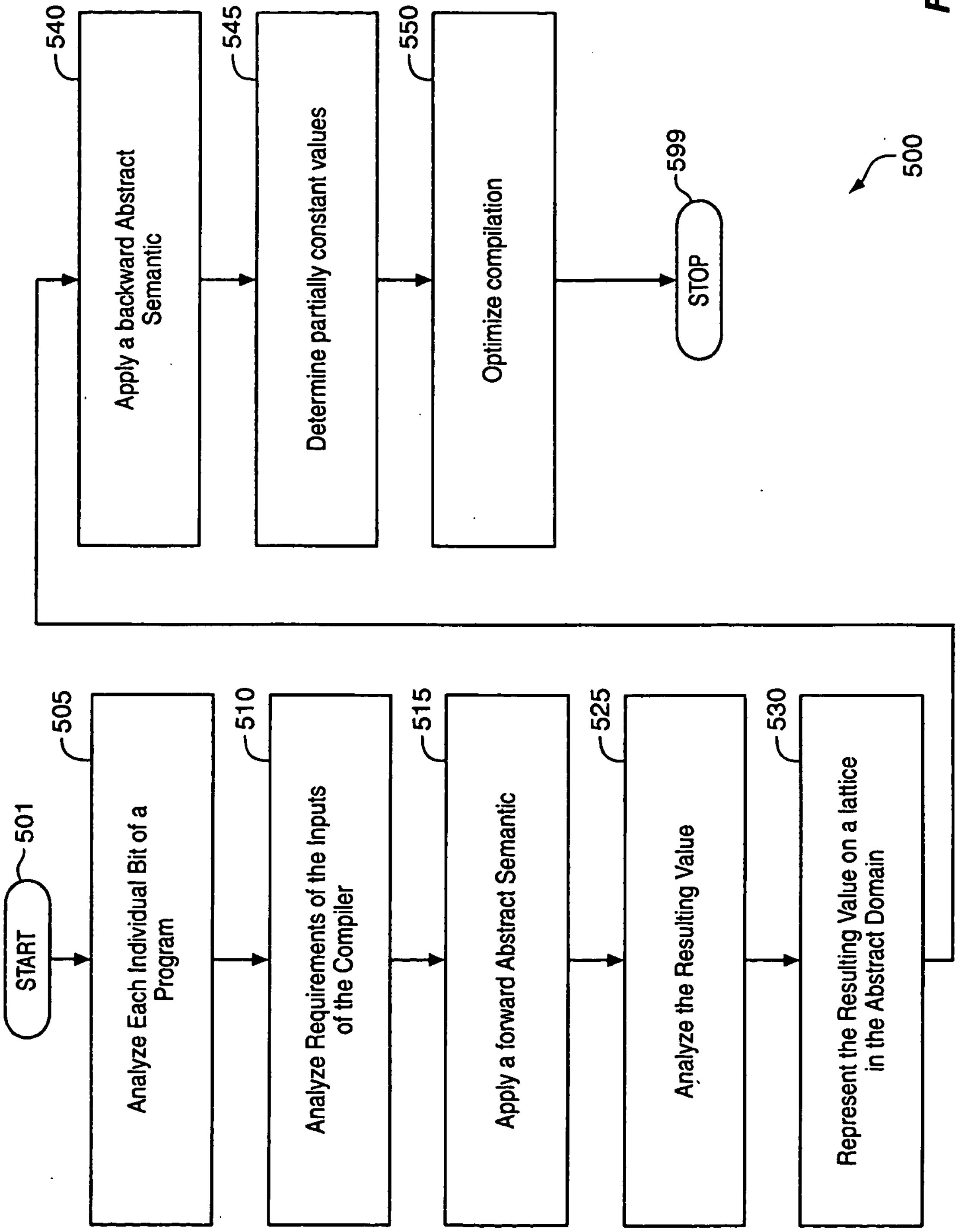


FIG. 5